

COD**E**BABA

NodeJS Blueprint

THIRD EDITION

Mastery official Codebook

ESLAM MOURAD

Nodejs curriculum



Copyright

NodeJs Blueprint : Mastery official curriculum

by Eslam Mourad

No Copyright © 2024 Eslam Mourad. All rights not reserved.

Published by <https://codebaba.net> Middle East, **Egypt**, Cairo.

Online editions are also available for most titles (codebaba.net)

Codebaba Curriculum may be purchased for free , this books educational for real world problem solving.

For more information, contact me at "<https://codebaba.net/contact-us/>"

Editor: Eslam Mourad

Production Editor: Eslam Mourad

Proofreader: Eslam Mourad

Indexer: Eslam Mourad

Cover Designer: Eslam Mourad

Interior Designer: Eslam Mourad

Illustrator: Eslam Mourad

ISBN: Non NaN

This is an educational curriculum that is suitable for all learning styles. You can subscribe to our community via social media through the official website. "<https://codebaba.net/contact-us/>"

Curriculum Overview

The JSNAD and JSNSD exams are two new tools created by the OpenJS Foundation that developers can use to demonstrate their Node.js skills, or pinpoint where they need to level up.

JSNAD: OpenJS Node Application Developer .

JSNSD: OpenJS Node Services Developer .

There are 2 certification exams.: <https://openjsf.org/certification/>

We going to take one, we focusing on the **JSNAD**.

<https://training.linuxfoundation.org/training/nodejs-application-development-lfw211/>

Knowing where to find great learning resources for each exam topic can be hard. Below is a list of tips, articles, tutorials, and videos we think can help.

What You learn:

- Demonstrate competence developing Node.js applications of any kind
- Focus on Node.js core APIs and common types of applications Node.js is used to build

Your environment:

Both exams are administered online and require using a Remote Desktop Linux environment. You'll be required to write code using either Visual Studio Code. You'll also be able to use Postman, Chrome, and the terminal. In some cases, partial code may be provided to save time. In other contexts, you'll write an entire program.

Outcome

We going to walk the path that turns things into results

Our result is that you are eligible for the job title "NodeJs Back-End Developer".

Basic

What is Node.js?

<https://nodejs.dev/en/learn/introduction-to-nodejs/>

<https://book.mixu.net/node/ch2.html>

Installing Node.js

Link <https://nodejs.org/en/download>

Link <https://nodejs.org/en/download/package-manager>

Link <https://nodejs.dev/en/learn/how-to-install-nodejs/>

Link <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>

Link <https://nodesource.com/blog/installing-node-js-tutorial-ubuntu/>

Buffers & Streams

Key competencies include; Node.js Buffer API's, incremental processing, transforming data, and connecting streams.

Node Cookbook, 4rd Edition - Chapter 3 & 4

Book <https://codebaba.net>

Buffer

Link https://en.wikipedia.org/wiki/Data_buffer

Api docs <https://nodejs.org/api/buffer.html>

Api docs <https://nodejs.org/dist/latest-v12.x/docs/api/buffer.html>

Link <https://nodejs.dev/en/api/v18/buffer/>

Link <https://nodesource.com/blog/understanding-the-buffer-deprecation-in-node-js-10/>

Streams

Link https://developer.mozilla.org/en-US/docs/Web/API/Streams_API

Link <https://www.freecodecamp.org/news/node-js-streams-everything-you-need-to-know-c9141306be93/>

api docs <https://nodejs.org/api/stream.html>

api docs <https://nodejs.org/dist/latest-v12.x/docs/api/stream.html>

api docs <https://nodejs.dev/en/api/v18/stream/>

Link <https://nodesource.com/blog/understanding-streams-in-nodejs/>

Streams Code

Link <https://github.com/dmitriz/stream-handbook>

Link <https://github.com/workshopper/stream-adventure>

Control flow

Key competencies include; Managing asynchronous operations and control flow abstractions.

What Are Promises?

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises

https://nodejs.org/dist/latest-v10.x/docs/api/fs.html#fs_fs_promises_api

<https://nodejs.dev/en/learn/reading-files-with-nodejs/>

Use Async/Await with Promises

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Promises>

<https://nodesource.com/blog/enterprise-grade-node-js-promises-with-async-and-bluebird/>

Control Flow

https://developer.mozilla.org/en-US/docs/Glossary/Control_flow

<https://nodejs.dev/en/learn/asynchronous-flow-control/>

<https://book.mixu.net/node/ch7.html>

Control Flow and error handling

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling

Child processes

Key competencies include; Spawning or executing child processes, and child process configuration.

Child_process documentation

api docs https://nodejs.org/api/child_process.html

api docs https://nodejs.org/docs/latest-v12.x/api/child_process.html

Node.js Child Processes: Everything you need to know

Link <https://www.freecodecamp.org/news/node-js-child-processes-everything-you-need-to-know-e69498fe970a/>

How to spawn a child process with Node.js

Link <https://flaviocopes.com/how-to-spawn-child-process-node/>

Debugging Node.js

Key competencies include; Debugging Node.js, basic performance analysis.

Node Cookbook, 3rd Edition - Chapter 1 - Debugging Processes

Book <https://codebaba.net>

Node Cookbook, 3rd Edition - Chapter 9 - Optimizing Performance

Book <https://codebaba.net>

[Node.js debugging in VS Code](#)

<https://code.visualstudio.com>

[Getting started with Node.js debugging in VS Code](#)

<https://www.youtube.com>

[Debugging Guide](#)

Link <https://nodejs.org>

[Easy profiling for Node.js Applications](#)

Link <https://nodejs.org>

[console documentation](#)

api docs <https://nodejs.org>

[debugger documentation](#)

api docs <https://nodejs.org>

[npmjs.com/package/0x](#)

Link <https://www.npmjs.com/package/0x>

Error handling

Key competencies include; Common error handling patterns, handling errors in various scenarios.

Better Error Handling In NodeJS With Error Classes**content**

<https://www.smashingmagazine.com/>

Handling Errors in JavaScript: The Definitive Guide

<https://levelup.gitconnected.com/>

Unhandled Promise Rejections in Node.js

<https://thecodebarbarian.com/>

How to prevent your Node.js process from crashing

<https://medium.com/>

Error documentation

api docs <https://nodejs.org/dist/latest-v18.x/docs/api/errors.html>

Node.js CLI

Key competencies include; Understand and use node executable command line flags.

Run ``man node`` in your CLI to get a complete list of available options.

Research and Practice

[Mastering the Node.js CLI & Command Line Options](https://blog.risingstack.com/mastering-the-nodejs-cli-command-line-options)

<https://blog.risingstack.com>

[Node.js CLI documentation](https://nodejs.org/en/docs/cli/)

api docs <https://nodejs.org>

Events

Key competencies include; Understand the event system, build event emitters, consume event emitters.

[How to code your own event emitter in Node.js: a step-by-step guide](#)

<https://www.freecodecamp.org/>

[Handling and dispatching events with Node.js](#)

<https://blog.logrocket.com/>

[Events documentation](#)

api docs <https://nodejs.org>

JavaScript prerequisites

Key competencies include; Language fundamentals, scoped to core language features introduced since EcmaScript 1 - or <https://tc39.es/ecma262/> - and still heavily used today.

Most Recommended JavaScript Learning Resources

<https://archive.org/details/javascript-official-content-codebaba.net-curriculum>

Via <https://codebaba.net>

Module system

Key competencies include; CommonJS Module System only (ES modules not introduced to Node core until after 10.x).

[Overview: Node's Module System](https://www.freecodecamp.org)

<https://www.freecodecamp.org>

[Requiring modules in Node.js: Everything you need to know](https://www.freecodecamp.org)

<https://www.freecodecamp.org>

Node Cookbook, 4rd Edition - Chapter 2- Writing Modules

Book <https://codebaba.net>

[module documentation](#)

api docs <https://nodejs.org>

Process/Operating system

Key competencies include; Controlling the process, and getting system data.

[Mastering the Node.js Core Modules - The Process Module](#)

<https://blog.risingstack.com/>

Node Cookbook, 3rd Edition - Chapter 3 - Coordinating IO

Book <https://codebaba.net>

[os documentation](#)

api docs <https://nodejs.org>

[process documentation](#)

api docs <https://nodejs.org>

Package.json

Key competencies include; Package configuration, and dependency management.

[What Is package.json?](#)

<https://docs.npmjs.com/>

[Create a package.json File](#)

<https://docs.npmjs.com>

[Install NPM Packages](#)

<https://docs.npmjs.com>

[What Is Semantic Versioning \(SemVer\)?](#)

<https://semver.org/>

[About semantic versioning](#)

<https://docs.npmjs.com>

[npm SemVer Calculator](#)

<https://semver.npmjs.com/>

[What Are NPM Scripts?](#)

<https://docs.npmjs.com>

Node Cookbook, 3rd Edition - Chapter 2- Writing Modules

Book <https://codebaba.net>

Unit testing

Key competencies include; Using assertions, testing synchronous code, and testing asynchronous code.
The exam and real work doesn't prescribe a specific framework;
We recommend you **pick one** and learn it.

Using assertions

<https://nodejs.org/dist/latest-v12.x/docs/api/assert.html>

<https://nodejs.dev/en/api/v19/assert/>

<https://nodejs.dev/en/api/v20/assert/>

SYNCHRONOUS CODE

<https://mochajs.org/#synchronous-code>

ASYNCHRONOUS CODE

<https://mochajs.org/#asynchronous-hooks>

<https://jestjs.io/docs/asynchronous>

Express.js

[Express/Node introduction](#)

In this first Express article we answer the questions "What is Node?" and "What is Express?" and give you an overview of what makes the Express web framework special. We'll outline the main features and show you some of the main building blocks of an Express application (although at this point you won't yet have a development environment in which to test it).

[Setting up a Node \(Express\) development environment](#)

Now that you know what Express is for, we'll show you how to set up and test a Node/Express development environment on Windows, Linux (Ubuntu), and macOS. Whatever common operating system you are using, this article should give you what you need to be able to start developing Express apps.

[Express Tutorial: The Local Library website](#)

The first article in our practical tutorial series explains what you'll learn and provides an overview of the "local library" example website we'll be working through and evolving in subsequent articles.

[Express Tutorial Part 2: Creating a skeleton website](#)

This article shows how you can create a "skeleton" website project, which you can then go on to populate with site-specific routes, templates/views, and databases.

[Express Tutorial Part 3: Using a Database \(with Mongoose\)](#)

This article briefly introduces databases for Node/Express. It then goes on to show how we can use [Mongoose](#) to provide database access for the *LocalLibrary* website. It explains how object schema and models are declared, the main field types, and basic validation. It also briefly shows a few of the main ways you can access model data.

[Express Tutorial Part 4: Routes and controllers](#)

In this tutorial we'll set up routes (URL handling code) with "dummy" handler functions for all the resource endpoints that we'll eventually need in the *LocalLibrary* website. On completion, we'll have a modular structure for our route handling code, that we can extend with real handler functions in the following articles. We'll also have a really good understanding of how to create modular routes using Express.

Express Tutorial Part 5: Displaying library data

We're now ready to add the pages that display the *LocalLibrary* website books and other data. The pages will include a home page that shows how many records we have of each model type and list and detail pages for all of our models. Along the way, we'll gain practical experience in getting records from the database and using templates.

Express Tutorial Part 6: Working with forms

In this tutorial we'll show you how to work with [HTML Forms](#) in Express, using Pug, and in particular how to write forms to create, update, and delete documents from the database.

Express Tutorial Part 7: Deploying to production

Now you've created an awesome *LocalLibrary* website, you're going to want to install it on a public web server so that it can be accessed by library staff and members over the Internet. This article provides an overview of how you might go about finding a host to deploy your website, and what you need to do in order to get your site ready for production.